Information and Computation 207 (2009)1209–1220

Contents lists available at ScienceDirect

Information and Computation

journal homepage: www.elsevier.com/locate/ic

Minimality in template-guided recombination☆

Michael Domaratzki

Department of Computer Science, University of Manitoba, Winnipeg, Man., Canada R3T 2N2

ARTICLE INFO

Article history:
Received 23 May 2008
Available online 26 March 2009

Keywords:
Ciliates
Template-guided recombination
DNA rearrangement
Equivalence
Minimality
Natural computing

ABSTRACT

Ciliates are unicellular organisms, some of which perform complicated rearrangements of their DNA. Template-guided recombination (TGR) is a formal model for the DNA recombination which occurs in ciliates. TGR has been the subject of much research in formal language theory, as it can be viewed as an operation on formal languages. In TGR, a set of templates serves as a parameter to a language operation which controls which rearrangements can take place; thus, a set of templates is itself a language.

Recently, the concept of equivalence in TGR has been considered: given two sets of templates, do they define the same language operation? This paper considers the related question of minimality: given a set of templates T, what is the smallest set of templates (with respect to inclusion) equivalent to T? We show that the minimal set of templates is unique, and consider closure properties and decidability questions related to minimality. We define an operational characterization for equivalence which is useful for results on minimality.

1. Introduction

Ciliates are unicellular eukaryotes, some of which have a well-studied ability to rearrange their DNA during conjugation, a form of reproduction. Initially, the scrambled genetic material is interspersed with internally eliminated segments (IESs), and may additionally be out of order. These segments of the scrambled gene which also constitute the final, unscrambled gene are called macronuclear destined segments (MDSs).

Formal models of this rearrangement are of interest to researchers in the field of natural computing, which is the field of computer science interested in computation outside of traditional silicon-based hardware. Several models for the DNA rearrangement in ciliates have been proposed; see for example Ehrenfeucht et al. [11] or Prescott et al. [20] for a description of ciliate DNA rearrangement and two formal models. There has been much recent work on various computational aspects of these different models of ciliate DNA rearrangement, including for example, decision problems for scrambled ciliate genes [18] and solutions for particular computational problems using gene assembly [1].

In this paper, we focus on template-guided recombination (TGR) as a formal model for ciliate DNA rearrangement. The TGR model, originally proposed by Prescott et al. [20], has been investigated in a language-theoretic framework by Daley and McQuillan [6–8], McQuillan et al. [16] and Daley et al. [5]. Most of these results focus on the closure properties of TGR as a language-theoretic operation. Recently, Angeleska et al. [2] have re-examined the TGR model and proposed the use of RNA templates in the TGR process. For a survey of TGR and related work, see Daley and the author [4].

The formal language model of TGR functions by providing a set of templates which defines the recombination events which are permitted to occur during DNA unscrambling (see Section 2 for definitions). Thus, the set of templates is itself

☆ Research supported in part by NSERC.
E-mail address: mdomarat@cs.umanitoba.ca (M. Domaratzki)

0890-5401/$ - see front matter © 2009 Elsevier Inc. All rights reserved
doi:10.1016/j.ic.2009.02.009

a formal language, and we can view the set of templates as an adjustable parameter of the TGR operation. With this view, the author has recently studied a language-theoretic concept of equivalence for sets of templates [10], where two sets of templates are equivalent if they define the same formal language operation.

Recent results by Nowacki et al. [17] demonstrate that the concept of templates is consistent with experimental observations. In particular, they show that the hypotheses of Prescott et al. [20] and Angeleska et al. [2] on the use of pre-existing genetic material as the set of templates in TGR appears to be supported by experimental results. By modifying the genetic material present during conjugation, Nowacki et al. [17] were able to alter the outcome of the rearrangement process. This suggests that it may be possible to control DNA rearrangement in ciliates.

In this paper, we use the concept of equivalence to investigate minimality of sets of templates. That is, for a given set of templates T, we are interested in the minimal set of templates (with respect to inclusion) which is equivalent to T. We show that this minimal set of templates is unique.

We demonstrate the closure properties of the operation which takes a set of templates to its minimal equivalent set of templates, and the decidability of whether a given set of templates is minimal. In particular, given a regular set of templates, we can construct its minimal set of templates (which is regular), and therefore also determine if it is minimal. For context-free sets of templates, it is undecidable whether a given set of templates is minimal.

2. Preliminaries

We use the tools of formal language theory to study TGR. For additional background on formal languages, see Rozenberg and Salomaa [21]. Let Σ be a finite set of symbols, called *letters*; we call Σ an *alphabet*. Then Σ^* is the set of all finite sequences of letters from Σ, which are called *words*. The empty word ε is the empty sequence of letters. We denote by Σ^+ the set of non-empty words over Σ, i.e., $\Sigma^+ = \Sigma^* - \{\varepsilon\}$. The length of a word w is denoted by $|w|$.

A word $x \in \Sigma^*$ is a *prefix* of a word $y \in \Sigma^*$ if there exists $w \in \Sigma^*$ such that $y = xw$. Similarly, x is a *suffix* of y if there exists $u \in \Sigma^*$ such that $y = ux$. If $x \in \Sigma^*$, then $\mathrm{pref}(x)$ (resp., $\mathrm{suff}(x)$) is the set of all prefixes (resp., suffixes) of x.

A *language* L is any subset of Σ^*. Given an alphabet Σ, we use the notation Σ^k to denote the set of all words in Σ^* of length k, while $\Sigma^{\geq k}$ (resp., $\Sigma^{\leq k}$) denotes the set of all words in Σ^* of length k or greater (resp., length k or less).

We assume the reader is familiar with the classes of regular and context-free languages. In particular, a language is *regular* (resp., *context-free*) if it is accepted (resp., generated) by a deterministic finite automaton (DFA) (resp., context-free grammar). The class of regular languages is strictly included in the class of context-free languages. Unless otherwise stated, we assume that all regular and context-free languages below are effectively given in what follows.

2.1. Template-guided recombination

We now give the formal definition of TGR, which was proposed by Prescott et al. [20] and first studied as a formal operation by Daley and McQuillan [6]. If $n_1, n_2 \geq 1$ and $x, y, z, t \in \Sigma^*$ are words, we denote by $(x, y) \vdash_{t,n_1,n_2} z$ the fact that we can write

$$x = u_1\alpha\beta v_1 \tag{1}$$
$$y = v_2\beta\gamma u_2 \tag{2}$$
$$z = u_1\alpha\beta\gamma u_2 \tag{3}$$
$$t = \alpha\beta\gamma \tag{4}$$

with $\alpha, \beta, \gamma, u_1, u_2, v_1, v_2 \in \Sigma^*$, $|\alpha|, |\gamma| \geq n_1$ and $|\beta| = n_2$. If n_1, n_2 are understood, then we denote the relation $\vdash_{t,n_1,n_2}$ by $\vdash_t$. The word t is called the *template*.

Example 1. Let $n_1, n_2 = 1$, $t_1 = aabc$, $t_2 = aab$ and $t_3 = abc$. Then we have

$$(caab, bc) \vdash_{t_1} caabc, \qquad (aa, abcd) \vdash_{t_2} aabcd,$$
$$(aa, abcd) \vdash_{t_1} aabcd, \qquad (caab, bc) \vdash_{t_3} caabc.$$

Intuitively, the words x and y represent the DNA strands which contain MDSs and which are to be recombined using the template t. The regions v_1 and v_2 represent the internal eliminated sequences (IESs) which do not form part of the final rearranged sequence, and β, which has a minimum length restriction, represents the pointer sequences in the ciliate DNA. The regions α and γ represent additional material which must flank the pointer sequence in the MDSs $u_1\alpha$ and γu_2. This representation leads to the asymmetry between x and y as input words. Note that in the definition of $(x, y) \vdash_t z$, the words x and y are separate DNA sequences and so TGR is an inter-molecular model for ciliate DNA recombination. More recently, however, an intra-molecular TGR has also been considered [5].

If $T, L \subseteq \Sigma^*$ are languages, then $\Uparrow_{T,n_1,n_2}(L)$ is defined by

$$\Uparrow_{T,n_1,n_2}(L) = \{z \,:\, \exists x, y \in L, t \in T \text{ such that } (x, y) \vdash_{t,n_1,n_2} z\}.$$

Again, we use the notation $\pitchfork_T(L)$ if n_1, n_2 are understood. The language T is the *set of templates*. In the TGR model, T represents the set of genetic material which serves to control rearrangement, and L represents the DNA being rearranged.

We note that the operation of TGR on words can be represented using Post canonical systems [19], which are rewriting systems with productions of the form

$$g_0P_1g_1P_2g_2\cdots g_{k-1}P_kg_k \to h_0P_{i_1}h_1P_{i_2}h_2\cdots h_{r-1}P_{i_r}h_r$$

where $g_0,\ldots,g_k,h_0,\ldots,h_r \in \Sigma^*$ and $P_1,P_2,\ldots,P_k$ are variables. The condition that $\{i_1,i_2,\ldots,i_r\} \subseteq \{1,\ldots,k\}$ is also enforced [19].

In particular, for a given decomposition $t=\alpha\beta\gamma$ with $|\alpha|,|\gamma| \geq n_1$ and $|\beta| = n_2$, let $R(y) = \mathrm{suff}(y) \cap \{\beta\gamma\}\Sigma^*$. We can then define the Post canonical system with rules

$$\{U\alpha\beta V \to U\alpha r \ : \ r \in R(y)\}.$$

Let $C(\alpha,\beta,\gamma)$ be this Post canonical system. Then we have that $x \Rightarrow z$ in $C(\alpha,\beta,\gamma)$ if and only if $(x,y) \vdash_{t,n_1,n_2} z$ for this particular decomposition $t=\alpha\beta\gamma$. The derivation relation $\Rightarrow^*$ of these Post canonical systems $C(\alpha,\beta,\gamma)$ are particularly suited to modelling iterated TGR (see, e.g., Daley and McQuillan [6]). We note, however, that in general this construction might not lead to a Post canonical system with a finite number of productions if the set of templates T is infinite or if the language L is infinite.

2.2. Equivalence and minimality

We now come to the definition of equivalence for sets of templates.

Definition 1. Let $n_1,n_2 \geq 1$. For $T_1,T_2 \subseteq \Sigma^*$, we say that T_1 and T_2 are (n_1,n_2)-*equivalent*, denoted by $T_1 \equiv_{n_1,n_2} T_2$, if $\pitchfork_{T_1,n_1,n_2}(L) = \pitchfork_{T_2,n_1,n_2}(L)$ for all $L \subseteq \Sigma^*$. By $T_1 \sqsubseteq_{n_1,n_2} T_2$, we mean $\pitchfork_{T_1,n_1,n_2}(L) \subseteq \pitchfork_{T_2,n_1,n_2}(L)$ for all languages $L \subseteq \Sigma^*$.

Note that $T_1 \equiv_{n_1,n_2} T_2$ if and only if $T_1 \sqsubseteq_{n_1,n_2} T_2$ and $T_2 \sqsubseteq_{n_1,n_2} T_1$.

Example 2. Let $T_1 = \{aabc, aab, abc\}, T_2 = \{aab, abc\}$. As $T_2 \subseteq T_1$, it immediately follows that $T_2 \sqsubseteq_{1,1} T_1$. However, and as we will see with Theorem 1 below, it is not hard to see that $T_1 \sqsubseteq_{1,1} T_2$.

In particular (see also Example 1), if $t = aabc \in T_1$ is factorized as $\alpha = a$, $\beta = a$ and $\gamma = bc$, then we can substitute its use by the use of the template $aab \in T_2$: the extra symbol c on the end of t is not crucial to the execution of the TGR operation in this case. On the other hand, the factorization $\alpha = aa$, $\beta = b$ and $\gamma = c$ is handled by the template $abc \in T_2$.

Thus, equivalence for sets of templates is a non-trivial property. Let (C1) be the following condition:

$$\begin{aligned} &\forall t,t_1,t_2 \in \Sigma^* \text{ with } |t| = 2n_1+n_2, \\ &\qquad \text{if } t_1tt_2 \in T_1 \text{ then } \exists t_1' \in \mathrm{suff}(t_1) \text{ and } \exists t_2' \in \mathrm{pref}(t_2)(t_1'tt_2' \in T_2). \end{aligned} \tag{C1}$$

Condition (C1) is illustrated in Fig. 1: for every subword t of length $2n_1+n_2$ in a template in T_1, there must be an extension of t in T_2 which agrees with the template in T_1 on the subwords flanking t.

Then we can give an exact characterization of equivalence, proven previously by the author [10].

Theorem 1. *Let Σ be an alphabet with $|\Sigma| \geq 3, n_1,n_2 \geq 1$ and $T_1,T_2 \subseteq \Sigma^*$. The condition (C1) holds if and only if $T_1 \sqsubseteq_{n_1,n_2} T_2$.*

That is, if T_1 is to be able to be replaced by T_2, we must have that every subword t of length $2n_1+n_2$ in T_1 must be able to be extended to a template in T_2 which agrees with the subwords which flank t as a subword in T_1.

We note that the condition that $|\Sigma| \geq 3$ in Theorem 1 is not known to be necessary. It is open whether the condition (C1) characterizes equivalence for alphabets of size two, or whether another equivalence condition applies to alphabets of size two. In what follows, the condition $|\Sigma| \geq 3$ will be inherited by results on minimality and equivalence which depend on Theorem 1 and its current proof.

In what follows, we will use the following terminology to describe this extendability property of templates:

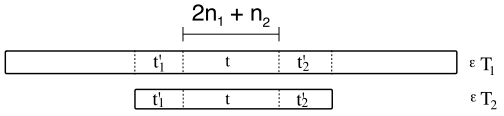

Fig. 1. Illustration of condition (C1).

Definition 2. Let $T \subseteq \Sigma^*$ and $t_1, t, t_2 \in \Sigma^*$ with $|t| = 2n_1 + n_2$. If $t' = t_1 t t_2$ is a template, then we say that the subword t of t' is *covered* by T if there exist subwords $t_1' \in \text{suff}(t_1), t_2' \in \text{pref}(t_2)$ such that $t_1' t t_2' \in T$.

We note in passing another recent use of the term *cover* in the context of ciliate DNA rearrangement under the template-guided model. Ehrenfeucht and Rozenberg [12] define the notion of covering on words which are defined as functions from an interval $[n_1, n_2]$ of integers to Σ; the intervals do not necessarily start at zero. This concept of coverings is motivated by the covering of the unscrambled macronuclear genes by the MDSs of the scrambled micronuclear gene. Due to the presence of pointers in the micronuclear gene, these MDSs overlap in the final macronuclear gene, and so the MDSs cover (with overlap) the macronuclear gene; see Ehrenfeucht and Rozenberg [12] for more details.

The motivation for covering in our context is therefore different from that of Ehrenfeucht and Rozenberg, and is with respect to ordinary words only. Further, we are mostly interested with the use of coverings for closure properties and decidability, whereas the results of Ehrenfeucht and Rozenberg focus on deep results on coverings in terms of cardinality and containment.

Finally, we define minimality for sets of templates, which will be the main focus of this paper.

Definition 3. Let $n_1, n_2 \geq 1$ and $T \subseteq \Sigma^*$. We say that a set of templates $T' \subseteq \Sigma^*$ is an (n_1, n_2)-minimal set of templates for T if $T' \equiv_{n_1,n_2} T$ and, for all $T'' \subsetneq T'$, $T'' \not\equiv_{n_1,n_2} T$.

That is, an (n_1, n_2)-minimal set of templates for T is a set of templates which is equivalent to T and which is minimal with respect to usual set inclusion. Note that, by definition, any (n_1, n_2)-minimal set of templates for T is a subset of T. We also say that T is (n_1, n_2)-minimal if T is an (n_1, n_2)-minimal set of templates for itself.

3. Operational characterization of equivalence

In this section, we give an additional formulation of Theorem 1 for equivalence of sets of templates, which will be useful for results on minimality. The alternate characterization is based on operations on formal languages. For any $N \geq 1$, let $\odot_N$ be the binary operation on words defined by

$$x \odot_N y = \{x_1 u y_1 \ : \ \exists x_1, y_1, u \in \Sigma^*, x = x_1 u, y = u y_1, |u| \geq N\}.$$

Note that if no prefix of y (of length N or more) is a suffix of x, then $x \odot_N y = \emptyset$. The operation is extended naturally to languages by $L_1 \odot_N L_2 = \{x \odot_N y \ : \ x \in L_1, y \in L_2\}$.

This operation is related to the Latin product of languages [15,14], where the overlap has length exactly one: i.e., the operation $u \bowtie v = \{u' a v' \ : \ u = u' a, v = a v'\}$.

Recent work by Ito and Lischke [13] defines another related operation $\otimes$, given by

$$p \otimes q = \{uvw \ : uv = p, vw = q, uw \neq \epsilon\}.$$

This is generalized to $p^{\otimes n}$ in the natural way: $p^{\otimes 0} = \{\epsilon\}$ and $p^{\otimes n} = \{w \otimes p : w \in p^{\otimes (n-1)}\}$. Ito and Lischke are concerned primarily in generalizations of primitive words using the operation $\otimes$. For instance, the authors define *hyper-periodic* words [13] as those words w such that $w \notin v^{\otimes n}$ for any $v \in \Sigma^*$ and any $n \geq 2$.

The operation $\odot_N$ is also similar to the concept of short concatenation defined by Cărăuşu and Păun [3]; the only difference in the definitions is that short concatenation takes the maximal overlap between x and y, if any. If no overlap occurs, then short concatenation of two words is the standard concatenation operation.

We also note that $\odot_N$ is defined by the semantic set of trajectories $S = 0^* \sigma^{\geq N} 1^*$ in the context of semantic shuffle on trajectories; see the author [9] for details. Informally, we can view the semantic set of trajectories as specifying the order of interleaving of symbols of the left and right input words: 0 (resp., 1) specifies that the next symbol of the result should be obtained from the left (resp., right) argument, while σ specifies that the next symbols of the left and right arguments must agree and only one copy of these two symbols is inserted into the resulting output.

From the results on semantic shuffle on trajectories [9], we can immediately conclude, e.g., that the regular languages are closed under $\odot_N$, and that if one of L_1, L_2 is context-free and the other is regular, then $L_1 \odot_N L_2$ is a CFL. This is in contrast to the case of short concatenation: the regular languages are not closed under short concatenation [3].

We also define an iterated version which will be more useful to us. Let

$$\odot_N^i(L) = \begin{cases} L & i = 0, \\ \odot_N^{i-1}(L) \odot_N L & i \geq 1. \end{cases}$$

$$\odot_N^*(L) = \bigcup_{i \geq 0} \odot_N^i(L).$$

$$\odot_N^{\leq i}(L) = \bigcup_{j \leq i} \odot_N^j(L).$$

The following example demonstrates the use of $\odot_N$ and $\odot_N^*$, and gives the intuitive idea of Lemma 8 below.

Example 3. For $N = 2$, consider that $aab \odot_N abc = \{aabc\}$. Thus, continuing with Example 2, if $T_2 = \{abc, aab\}$ we get that $\odot_N^*(T_2) = \{abc, aab, aabc\} = T_1$.

For completeness, we briefly investigate the associativity of $\odot$. We can see that, in general, $\odot$ is not associative. However, when considering $\odot^*$, we show that we can ignore the nonassociativity of the operation.

Let $\overline{\odot}_N^i$, $\overline{\odot}_N^*$ and $\overline{\odot}_N^{\leq i}$ be the operations defined by

$$\overline{\odot}_N^i(L) = \begin{cases} L & i = 0, \\ L \odot_N \overline{\odot}_N^{i-1}(L) & i \geq 1. \end{cases}$$

$$\overline{\odot}_N^{\leq i}(L) = \bigcup_{j \leq i} \overline{\odot}_N^j(L).$$

$$\overline{\odot}_N^*(L) = \bigcup_{i \geq 0} \overline{\odot}_N^i(L).$$

We can easily verify the following identities:

$$(L_1^R \odot_N L_2^R)^R = L_2 \odot_N L_1, \tag{5}$$

$$\overline{\odot}_N^i(L^R) = (\odot_N^i(L))^R \quad \forall i \geq 1. \tag{6}$$

Lemma 2. *For all $L \subseteq \Sigma^*$ and all integers $N, i \geq 0$, we have $\overline{\odot}_N^{\leq i}(L) = \odot_N^{\leq i}(L)$.*

Proof. Let L be an arbitrary language and $N \geq 0$. The proof is by induction on i. For $i = 0$, both the left- and right-hand side of the equality are L. For $i = 1$, both are equal to $L \odot_N L$.

Let $i \geq 2$. Assume that $\overline{\odot}_N^{\leq i-1}(L) = \odot_N^{\leq i-1}(L)$. We now show that $\overline{\odot}_N^{\leq i}(L) = \odot_N^{\leq i}(L)$.

Consider that

$$\odot_N^i(L) = (\odot_N^{i-1}(L)) \odot L \subseteq (\overline{\odot}_N^{\leq i-1}(L)) \odot L.$$

Thus, let $x \in \odot_N^i(L)$. Then there exists some $j < i$, $y \in \overline{\odot}_N^j(L)$ and $z \in L$ such that $x \in y \odot_N z$.

Consider that if $j = 0$, then $y \in L$ and $x \in L \odot_N L \subseteq \overline{\odot}_N^{\leq i}(L)$ (as $i \geq 2$). Thus, we may assume that $j \geq 1$. Therefore, we can write $y = u \odot_N v$ where $u \in L$ and $v \in \overline{\odot}_N^{j-1}(L) \subseteq \odot_N^{\leq j-1}(L)$. Let $j > k \geq 0$ be such that $v \in \odot_N^k(L)$.

Now write

$$x = \alpha\beta\gamma, \quad y = \alpha\beta, \quad z = \beta\gamma,$$

where $|\beta| \geq N$, as well as

$$y = \zeta\eta\theta, \quad u = \zeta\eta, \quad v = \eta\theta,$$

where $|\eta| \geq N$. We distinguish between two cases:

(a) $|v| \geq |\beta|$. Then as $v = \eta\theta$, and $y = \zeta\eta\theta = \alpha\beta$, we can write $\eta\theta = s\beta$ and $\alpha = \zeta s$ for some $s \in \Sigma^*$. But now, note that $z = \beta\gamma$ and $v = \eta\theta = s\beta$. As $|\beta| \geq N$, we then have that $s\beta\gamma \in v \odot_N z$. Thus, $s\beta\gamma \in \odot_N^{k+1}(L)$. Note that $k + 1 \leq j < i$. Thus, by induction $s\beta\gamma \in \overline{\odot}_N^{\ell}(L)$ for some $\ell \leq k + 1 < i$. Finally, $y = \alpha\beta = \zeta s\beta$, so that $x = \alpha\beta\gamma \in y \odot_N s\beta\gamma$. Thus $x \in \overline{\odot}_N^{\ell+1}(L)$. Note that $\ell + 1 \leq i$, so that $x \in \overline{\odot}_N^{\leq i}(L)$.

(b) $|v| < |\beta|$. As $y = \zeta\eta\theta = \alpha\beta$, in this case we can write

$$\beta = s\eta\theta,$$
$$\zeta = \alpha s.$$

Now, $z = \beta\gamma = s\eta\theta\gamma$ and $u = \zeta\eta = \alpha s\eta$. So as $|\eta| \geq N$, we have $u \odot_N z \ni \alpha s\eta\theta\gamma = \alpha\beta\gamma = x$. Thus, $x \in L \odot_N L \subseteq \overline{\odot}_N^{\leq i}(L)$ as $i \geq 2$.

Thus, for all $x \in \odot_N^i(L)$, we have $x \in \overline{\odot}_N^{\leq i}(L)$. We conclude that $\odot_N^{\leq i}(L) \subseteq \overline{\odot}_N^{\leq i}(L)$. The reverse inclusion follows immediately from (5). □

Corollary 3. *For all languages L, $\odot_N^*(L) = \overline{\odot}_N^*(L)$.*

Thus, in what follows, we do not consider issues of associativity with regards to words in $\odot_N^*(L)$. We now consider a positive closure property of $\odot_N^*$.

Theorem 4. *Let $L \subseteq \Sigma^*$ be a regular language and $N \geq 1$. Then $\odot_N^*(L)$ is a regular language.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. Recalling that 2-way NFAs accept only regular languages, we describe a 2-way NFA $M_\odot$ which accepts $\odot_N^*(L)$. The 2-way NFA $M_\odot$ will process inputs words with left and right endmarkers, and perform the following actions:

1. From the initial state q_0 of M, $M_\odot$ moves right, processing input in the same way that M does.
2. When a state from F is encountered, $M_\odot$ nondeterministically chooses to continue processing the input (i.e., ignore the current final state of M), begin moving left, or enter a final state and end processing if the next symbol is the right endmarker.
3. Once $M_\odot$ is moving left, it must do so for at least N steps (if it is less than N steps from the left end of the input, the computation is not successful). The machine may nondeterministically choose to move more than N steps to the left. The contents of the tape are ignored while $M_\odot$ is moving left.
4. After $M_\odot$ has chosen to stop moving left, it returns to the state q_0 of M and to step 1.

Thus, $M_\odot$ scans for a word from L, then backs up over the overlapping portion and starts to scan another occurrence of a word in L. The situation is depicted in Fig. 2.

We note that, as depicted in Fig. 2, it is not necessary that the positions of the start state q_0 form a sequence which progresses from left-to-right in the input word. It is possible for the 2-way NFA to sweep back further than a previous scan. In this case, the intermediate, swept-over scans of any words $x_1, x_2, \ldots, x_n \in L$ can be ignored. We can view the assembly of a word from $\odot_N^*(L)$ in this case as originating from all words scanned prior to x_1 as well as the word whose scan subsumed $x_1, \ldots, x_n$, and any subsequent words. (Note, of course, that some of these words may later be swept-over by future scans, and thus ignored.) This assembly of a word in $\odot_N^*(L)$ is possible since the overlaps are guaranteed to meet the required minimum length N.

Recalling that a 2-way NFA accepts if it enters a final state while at the right end of the input, we can see that the halting condition in step 2 does in fact recognize occurrences of words in $\odot_N^*(L)$. □

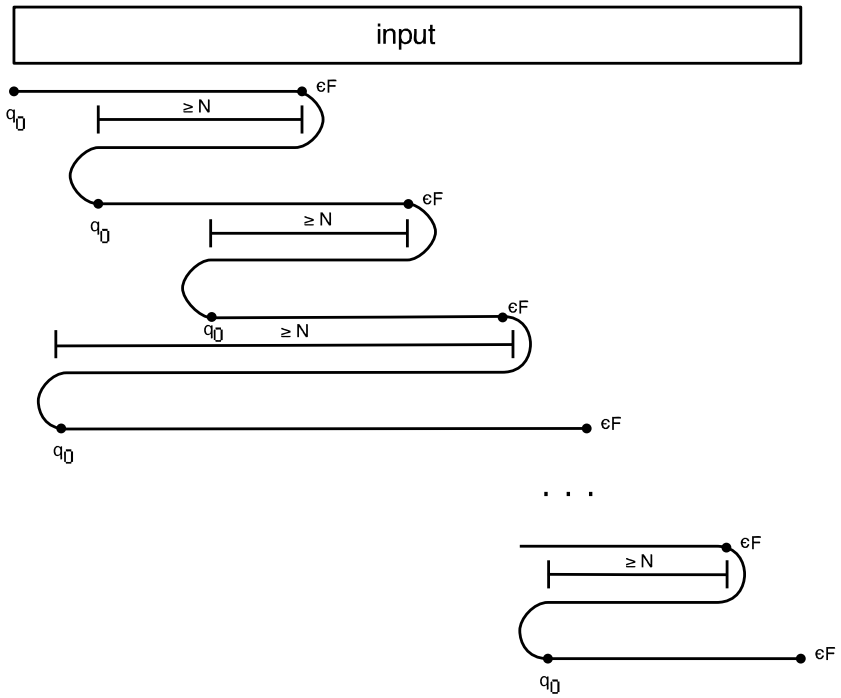

Fig. 2. An accepting computation of $M_\odot$ on a word in $\odot_N^*(L)$. Successive computations of M are overlapped by at least N symbols. Ultimately, the 2-way NFA $M_\odot$ reaches the right end of the input while in a final state of M, and accepts.

We also require a refinement of $\odot_N$, which is similar to the definition of Ito and Lischke [13], which will be useful for proving minimality results. For any $N \geq 1$, let $\otimes_N$ be the binary operation on words defined by

$$x \otimes_N y = \{x_1 u y_1 \ : \ \exists u \in \Sigma^*, x_1, y_1 \in \Sigma^+, |u| \geq N, x = x_1 u, y = u y_1\}.$$

Thus, the only difference between $\otimes_N$ and $\odot_N$ is that overlap between x and y cannot be all of x or all of y. Again, $L_1 \otimes_N L_2 = \{x \otimes_N y \ : \ x \in L_1, y \in L_2\}$ for all $L_1, L_2 \subseteq \Sigma^*$.

We again note that $\otimes_N$ is also defined by a regular semantic set of trajectories, which ensures the same closure properties as $\odot_N$. It is also interesting to note that $\otimes_N$ is very similar to a TGR operation. In particular, if $T_N = \Sigma^{N+2}$, then the operation $\pitchfork_{T_N,1,N}(\cdot)$ gives the proper overlap for an input language with itself, but as it also allows deletions of material (i.e., the words v_1 and v_2 in (1) and (2), respectively), we cannot immediately conclude, e.g., closure properties such as Theorem 6 below.

For iterated versions, let

$$\otimes_N^i(L) = \begin{cases} L & i = 0, \\ \otimes_N^{i-1}(L) \otimes_N L & i \geq 1. \end{cases}$$
$$\otimes_N^+(L) = \bigcup_{i \geq 1} \otimes_N^i(L).$$

Note that the definition of $\otimes_N^+$ explicitly removes L from the language, as $i \geq 1$ in the infinite union. We have the following useful observation:

Proposition 5. *If $x \in y \otimes_N z$, then $|x| > \max\{|y|, |z|\}$.*

The modification of $\otimes_N$ does not affect the closure properties of the iterated version for regular languages:

Theorem 6. *Let $L \subseteq \Sigma^*$ be a regular language and $N \geq 1$. Then $\otimes_N^+(L)$ is a regular language.*

Proof. Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA for L. We describe a 2-way NFA $M_\otimes$ with input endmarkers which accepts $\otimes_N^+(L)$. It is a modification of the construction of Theorem 4. The main distinction is the presence of a bit in the finite control of the NFA $M_\otimes$, which we call the *pass* bit. This bit will indicate whether the NFA is attempting to accept a word in $\otimes_N^+(L)$ via a single word of L.

1. Initially, the *pass* bit is set to 1.
2. From the initial state q_0 of M, $M_\otimes$ moves right, processing input in the same way that M does.
3. When a state from F is encountered, $M_\otimes$ nondeterministically chooses to continue processing the input or begin moving left.
4. Additionally, if a state from F is encountered, *pass* is 0, and the next input symbol is the right endmarker, $M_\otimes$ may nondeterministically choose to enter a final state and end computation.
5. If $M_\otimes$ chooses to move left, the *pass* bit is set to 0. Then, $M_\otimes$ must move left for at least N steps (if it is less than N steps from the left end of the input, the computation is not successful). The machine may nondeterministically choose to move more than N steps to the left. The contents of the tape are ignored while moving left.
6. If, while moving left, $M_\otimes$ reaches the left endmarker of the input, the *pass* bit is reset to 1.
7. After $M_\otimes$ has chosen to stop moving left, it returns to the state q_0 and to step 1.

The idea is similar to the proof of Theorem 4, except that, as illustrated in Fig. 3, we must not accept by starting at the left marker, completing a computation of M, and then entering a final state at the right marker. □

The following result, which states that the iterated $\odot_{2n_1+n_2-1}$ operation creates new templates all of whose subwords of length $2n_1 + n_2$ are covered, will be useful for establishing our main result relating $\odot_N$ to equivalence of templates.

Lemma 7. *Let $n_1, n_2 \geq 1$, $t \in \Sigma^*$ and $T \subseteq \Sigma^*$. If $t \in \odot_{2n_1+n_2-1}^*(T)$ then all subwords of t of length $2n_1 + n_2$ are covered by T.*

Proof. As $t \in \odot_{2n_1+n_2-1}^*(T)$, we have that $t \in \odot_{2n_1+n_2-1}^i(T)$ for some $i \geq 0$. The proof is by induction on i. For $i = 0$, $t \in T$. Then every subword of t is covered by t, which is in T.

Let $i \geq 1$ and assume that for all $j < i$ and all $t \in \odot_{2n_1+n_2-1}^j(T)$, all subwords of t of length $2n_1 + n_2$ are covered by T.

Let $t \in \odot_{2n_1+n_2-1}^i(T)$. Then by definition, $t = t_1 t_2 t_3$ where $t_1 t_2 \in \odot_{2n_1+n_2-1}^{i-1}(T)$, $t_2 t_3 \in T$ and $|t_2| \geq 2n_1 + n_2 - 1$. By induction, all subwords of length $2n_1 + n_2$ of $t_1 t_2$ are covered by T. Further, all subwords of $t_2 t_3$ are covered by $t_2 t_3$, which is in T. But as $|t_2| \geq 2n_1 + n_2 - 1$, all subwords of length $2n_1 + n_2$ are either a subword of $t_1 t_2$ or of $t_2 t_3$. □

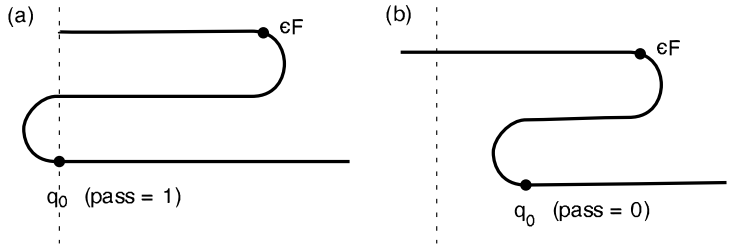

Fig. 3. Setting the *pass* bit: (a) if the left endmarker of the input (represented by the dotted vertical line) is reached while moving left, then *pass* is set to one. (b) Otherwise, we start new computations from q_0 with the pass bit set to zero.

We now describe our first result which motivates the introduction of the $\odot_N$ operation for studying TGR. The following lemma gives another characterization of $\sqsubseteq_{n_1,n_2}$ in terms of the operation $\odot_N^*$.

Lemma 8. *Let Σ be an alphabet with $|\Sigma| \geq 3$. For all $T_1, T_2 \subseteq \Sigma^*$ and all $n_1, n_2 \geq 1$, $T_1 \sqsubseteq_{n_1,n_2} T_2$ if and only if $T_1 \subseteq \odot^*_{2n_1+n_2-1}(T_2)$.*

Proof. The right-to-left implication holds by Lemma 7 and Theorem 1. For the reverse implication, let $t \in T_1$ and $T_1 \sqsubseteq_{n_1,n_2} T_2$. Consider the prefix t_0 of t of length $2n_1 + n_2$, and write $t = t_0 s_0$. By (C1), there is a template $t_0 r_0$ where $r_0 \in \text{pref}(s_0)$ and $t_0 r_0 \in T_2$.

If $r_0 = s_0$, then $t = t_0 s_0 \in T_2 \subseteq \odot^*_{2n_1+n_2-1}(T_2)$ and we are done. Otherwise, r_0 is a proper prefix of s_0 and we can write $t = t_0 r_0 a_0 s_1$ where $a_0 \in \Sigma$ and $s_1 \in \Sigma^*$. Note that $|s_0| > |s_1|$ and that $|t_0 r_0 a_0| > 2n_1 + n_2$. By the second fact, we can write $t = t_0' t_1 s_1$ where $|t_1| = 2n_1 + n_2$ and the final letter of t_1 is a_0.

Considering now the factorization $t = t_0' t_1 s_1$, by (C1) there is a template $u_0 t_1 r_1 \in T_2$ such that $u_0 \in \text{suff}(t_0')$ and $r_1 \in \text{pref}(s_1)$. Note that $t_0' t_1 r_1 \in t_0' t_1 \odot_{2n_1+n_2-1} u_0 t_1 r_1 \subseteq T_2 \odot_{2n_1+n_2-1} T_2$.

At this point again, we note that if $r_1 = s_1$, then we are done, as $t = t_0' t_1 r_1 \in T_2 \odot_{2n_1+n_2-1} T_2$. Otherwise, we can write $s_1 = r_1 a_1 s_2$ with $a_1 \in \Sigma$ and $s_2 \in \Sigma^*$ and repeat the above process. Since $|s_0| > |s_1| > |s_2|$, as we continue this process, it must eventually stop and we get that $t \in \odot^*_{2n_1+n_2-1}(T)$. □

Thus, Lemma 8 and Theorem 4 yield, for example, that given regular sets of templates T_1, T_2 it is decidable whether $T_1 \equiv_{n_1,n_2} T_2$ for some fixed $n_1, n_2 \geq 1$. This was previously established by the author using a pumping-type argument [10]; we note that neither immediately gives an efficient algorithm for testing equivalence of regular sets of templates given as DFAs. Determining lower bounds on the descriptional complexity of $\otimes_N^*(T)$ is open, as is determining the true computational complexity of determining if $T_1 \equiv_{n_1,n_2} T_2$.

4. Uniqueness and characterization of minimality

We now turn to (n_1, n_2)-minimal sets for a given set of templates. We show that the (n_1, n_2)-minimal set of templates for a given set of templates is unique. We consider this minimal set and give an explicit construction for it in terms of the operation $\otimes_N^+$.

The following lemma states that when T_1 and T_2 are (n_1, n_2)-equivalent, if we wish to cover a subword of a template in one of the two sets, we can always do so with a template which is in the intersection of T_1 and T_2.

Lemma 9. *Let $T_1, T_2 \subseteq \Sigma^*$ ($|\Sigma| \geq 3$) with $T_1 \equiv_{n_1,n_2} T_2$. Let $t \in T_1$ be such that $t = t_1 t_2 t_3$ with $|t_2| = 2n_1 + n_2$ for some $t_1, t_2, t_3 \in \Sigma^*$. Then there exists $t' \in T_1 \cap T_2$ such that $t' \in \text{suff}(t_1) t_2 \text{pref}(t_3)$.*

Note that Lemma 9 is true by Theorem 1 if the set $T_1 \cap T_2$ is replaced by T_2. Intuitively, Lemma 9 holds since the condition in (C1) maintains or decreases the length of templates, so repeated applications of the construction implied by (C1) must converge to a template whose length is at least $2n_1 + n_2$.

Proof. The proof is by induction on the length of t. The base case is t being a shortest template in T_1 of length at least $2n_1 + n_2$. Write $t = t_1 t_2 t_3$ where $|t_2| = 2n_1 + n_2$.

We claim that in this case, $t \in T_2$. If not, since $T_1 \equiv_{n_1,n_2} T_2$, there exists $t_0 \in T_2$ with $t_0 \in \text{suff}(t_1) t_2 \text{pref}(t_3)$ by Theorem 1. Thus $|t_0| \leq |t|$. In fact, $|t_0| < |t|$, since otherwise $t = t_0 \in T_2$, a contradiction to our assumption that $t \notin T_2$. Write $t_0 = t_1' t_2 t_3'$ where $t_1' \in \text{suff}(t_1)$ and $t_3' \in \text{pref}(t_3)$.

Now, $t_0 \in T_2 \equiv_{n_1,n_2} T_1$ so there exists $t_0' \in T_1$ such that $t_0' \in \text{suff}(t_1')t_2\text{pref}(t_3')$. Clearly,

$$2n_1 + n_2 \leq |t_0'| \leq |t_0| < |t|,$$

a contradiction to the choice of t as a shortest template in T_1 of length at least $2n_1 + n_2$. Thus, $t \in T_1 \cap T_2$, which establishes the base case.

Assume the lemma holds for all templates in T_1 with length at most k. Let $t \in T_1$ be a template of minimal length among all of those with length greater than k. Write $t = t_1t_2t_3$ with $|t_2| = 2n_1 + n_2$. As $T_1 \equiv_{n_1,n_2} T_2$, there exists $t' \in \text{suff}(t_1)t_2\text{pref}(t_3)$ such that $t' \in T_2$. Let $t' = t_1't_2t_3'$ where $t_1' \in \text{suff}(t_1)$ and $t_3' \in \text{pref}(t_3)$. Further, there exists $t'' \in T_1$ such that $t'' \in \text{suff}(t_1')t_2\text{pref}(t_3')$. Let $t'' = t_1''t_2t_3''$ where $t_1'' \in \text{suff}(t_1')$ and $t_3'' \in \text{pref}(t_3')$. Note that $|t''| \leq |t'| \leq |t|$. If $|t''| = |t|$, then $t'' = t' = t \in T_1 \cap T_2$ and we are done. Otherwise, $|t''| < |t|$ and by induction, there exists $s \in \text{suff}(t_1'')t_2\text{pref}(t_3'') \subseteq \text{suff}(t_1)t_2\text{pref}(t_3)$ such that $s \in T_1 \cap T_2$. Thus, s satisfies the conditions of the lemma, which now holds by induction. □

Theorem 10. *If $T_1, T_2 \subseteq \Sigma^*$ ($|\Sigma| \geq 3$) with $T_1 \equiv_{n_1,n_2} T_2$, then $T_1 \equiv_{n_1,n_2} T_1 \cap T_2$.*

Proof. Let $T_1, T_2 \subseteq \Sigma^*$ with $T_1 \equiv_{n_1,n_2} T_2$. As $T_1 \cap T_2 \subseteq T_1$, we have $\Cap_{T_1 \cap T_2}(L) \subseteq \Cap_{T_1}(L)$ for all $L \subseteq \Sigma^*$. Thus, $T_1 \cap T_2 \sqsubseteq T_1$. On the other hand, by Lemma 9 and Theorem 1, we have $T_1 \sqsubseteq T_1 \cap T_2$. □

Corollary 11. *Let $n_1, n_2 \geq 1$ and $T \subseteq \Sigma^*$ ($|\Sigma| \geq 3$). Then the (n_1, n_2)-minimal set for T is unique.*

Proof. If T_1, T_2 are both (n_1, n_2)-minimal sets of templates for T which are incomparable, then $T \equiv_{n_1,n_2} T_1 \cap T_2$, which contradicts the minimality of T_1 and T_2. □

In what follows, for a set T, we denote by $\beta_{n_1,n_2}(T)$ the (n_1, n_2)-minimal set of templates for T. Note that, by definition, we have that $\beta_{n_1,n_2}(T) \subseteq T$, since any (n_1, n_2)-minimal set of templates is obtained by removing templates from T. We begin with the following lemma on $\beta_{n_1,n_2}(T)$.

Lemma 12. *Let $T \subseteq \Sigma^*$ be a set of templates. If $t \in T - \beta_{n_1,n_2}(T)$, then for all factorizations of $t = t_1t_2t_3$ where $|t_2| = 2n_1 + n_2$, there exists $t' \in \beta_{n_1,n_2}(T)$ such that $t' \in \text{suff}(t_1)t_2\text{pref}(t_3)$.*

Proof. Note first that if $\beta_{n_1,n_2}(T) \subseteq T' \subseteq T$, then $T \equiv_{n_1,n_2} T' \equiv_{n_1,n_2} \beta_{n_1,n_2}(T)$. Thus, we have that $\beta_{n_1,n_2}(T) \cup \{t\} \equiv_{n_1,n_2} \beta_{n_1,n_2}(T)$, since $t \in T$. With this, the claim holds by applying Lemma 9 to the equivalent sets of templates $\beta_{n_1,n_2}(T)$ and $\beta_{n_1,n_2}(T) \cup \{t\}$, with the template of interest being t. □

The following theorem states that $\beta_{n_1,n_2}(T)$ consists of exactly those templates for which some subword of length $2n_1 + n_2$ cannot be extended to any other template in T.

Theorem 13. *Let Σ be an alphabet of size at least three. For all $T \subseteq \Sigma^*$,*

$$\begin{aligned}\beta_{n_1,n_2}(T) = \{t \in T : \exists t_0 \in \Sigma^{2n_1+n_2} \text{ and } t_1, t_2 \in \Sigma^* \\ \text{such that } t = t_1t_0t_2 \text{ and } T \cap \text{suff}(t_1)t_0\text{pref}(t_2) = \{t\}\}.\end{aligned} \tag{7}$$

Proof. Let $t \in \beta_{n_1,n_2}(T)$ but assume that for all factorizations $t = t_1t_0t_2$ with $|t_0| = 2n_1 + n_2$, there exists $t' \in \text{suff}(t_1)t_0\text{pref}(t_2) \cap T$ such that $t' \neq t$. If, for all factorizations of $t = t_1t_0t_2$, the constructed covering word t' is in $\beta_{n_1,n_2}(T)$ then we are done, as in this case $\beta_{n_1,n_2}(T) - \{t\} \equiv_{n_1,n_2} \beta_{n_1,n_2}(T)$, contradicting the minimality of $\beta_{n_1,n_2}(T)$. But on the other hand, if $t' \in T - \beta_{n_1,n_2}(T)$ for some factorization, then by Lemma 12 we have that there exists $t'' \in \beta_{n_1,n_2}(T)$ such that t'' is also in $\text{suff}(t_1)t_0\text{pref}(t_2)$. Thus, we have reduced this case to the previous one, again giving a contradiction.

For the reverse containment, let $t \notin \beta_{n_1,n_2}(T)$. If $t \notin T$, then clearly, t is not in the right-hand side of (7). Thus, assume that $t \in T - \beta_{n_1,n_2}(T)$. By Lemma 12, we immediately get that t is not in the right-hand side of (7). □

We now use the modified overlap operation $\otimes_N$ to characterize the (n_1, n_2)-minimal set of templates for a given set of templates T. We need $\otimes_N$ instead of $\odot_N$ as we need to remove only words which are constructed by a non-trivial overlapping.

Lemma 14. *Let Σ be an alphabet of size at least three. For all $n_1, n_2 \geq 1$ and $T \subseteq \Sigma^{\geq 2n_1+n_2}$, $\beta_{n_1,n_2}(T) = T - \otimes^+_{2n_1+n_2-1}(T)$.*

Proof. Let $t \notin T - \otimes^+_{2n_1+n_2-1}(T)$. If $t \notin T$, then $t \notin \beta_{n_1,n_2}(T)$. Thus, $t \in \otimes^+_{2n_1+n_2-1}(T)$. By Lemma 7, there are $t_1, t_2, \ldots, t_m \in T$, with $m \geq 2$, where every subword of t of length $2n_1 + n_2$ is covered by some t_i with $1 \leq i \leq m$. Note that $t \notin \{t_1, \ldots, t_m\}$, as $|t_i| < |t|$ for all $1 \leq i \leq m$ by Proposition 5. Thus, for every factorization of t as $t = s_1s_2s_3$ where $|s_2| = 2n_1 + n_2$, there exists some t_i such that $t_i \in T \cap \text{suff}(s_1)s_2\text{pref}(s_3)$. As $|t_i| < |t|$, $t \notin \beta_{n_1,n_2}(T)$ by Theorem 13.

Let $t \notin \beta_{n_1,n_2}(T)$. Note that if $t \notin T$, then clearly $t \notin T - \otimes^+_{2n_1n+n_2-1}(T)$. Thus, assume that $t \in T - \beta_{n_1,n_2}(T)$. Note that $|t| > 2n_1 + n_2$, since if $|t| = 2n_1 + n_2$, then $t \in \beta_{n_1,n_2}(T)$ by Lemma 12.

Since $t \in T - \beta_{n_1,n_2}(T)$, Theorem 13 states that for all subwords t' of t of length $2n_1 + n_2$, there exists a template s in T shorter than t such that t' is covered by s. Note that there is more than one subword of t of length $2n_1 + n_2$, as $|t| > 2n_1 + n_2$. By the same proof idea as Lemma 8, we can show that $t \in \otimes^+_{2n_1+n_2-1}(T)$. □

Example 4. Let $n_1 = n_2 = 1$ and $T = \{aabc, abcd, bcd, aabcd, cdd, aabcdd\}$. Then note that $aabcd \in aabc \otimes_2 bcd$ and $aabcdd \in aabcd \otimes_2 cdd$. We can verify that no other words of T may be obtained using $\otimes_2$, and so

$$\beta_{1,1}(T) = \{aabc, abcd, bcd, cdd\}.$$

Note that $abcd \in \beta_{1,1}(T)$ since even though $aabcd \notin \beta_{1,1}(T)$, $abcd$ itself cannot be directly constructed using $\otimes^+_2$.

5. Closure properties and decidability

We now consider closure properties of taking the minimal set of templates for given classes of languages, and the decidability of minimality for sets of templates. For regular languages, closure and decidability follow from the closure properties of $\otimes^+_N$.

Lemma 15. *For all $n_1, n_2 \geq 1$ and all $T \subseteq \Sigma^*$, $\beta_{n_1,n_2}(T)$ is a regular set of templates.*

Proof. By Lemma 14, it suffices to show that $T - \otimes^+_{2n_1+n_2-1}(T)$ is a regular language. But $\otimes^+_{2n_1+n_2-1}(T)$ is regular by Theorem 6, and the regular languages are closed under set difference. □

Lemma 16. *For all $n_1, n_2 \geq 1$, it is decidable if a regular set of templates T is (n_1, n_2)-minimal.*

Proof. By Lemma 15, $\beta_{n_1,n_2}(T)$ is regular since T is. Thus, we can test the equality $T = \beta_{n_1,n_2}(T)$. □

Lemma 17. *For all $n_1, n_2 \geq 1$ there exists a context-free set of templates T_{n_1,n_2} such that $\beta_{n_1,n_2}(T_{n_1,n_2})$ is not context-free.*

Proof. Let T_{n_1,n_2} be defined as

$$\begin{aligned} T_{n_1,n_2} = &\{\$a^i b^j c^j \# : i,j \geq 2n_1 + n_2\} \cup \{\$a^i b^i \ : \ i \geq 2n_1 + n_2\} \\ &\cup \{b^i c^j \ : \ i,j \geq 0, i + j = 2n_1 + n_2\} \cup \{c^{2n_1+n_2-1}\#\}. \end{aligned}$$

We claim that

$$\beta_{n_1,n_2}(T_{n_1,n_2}) \cap \$a^* b^* c^* \# = \{\$a^i b^j c^j \# \ : \ i \geq j \geq 2n_1 + n_2\}. \tag{8}$$

It is easy to see that the right side of (8) is not a context-free language. To verify the equality (8), note that $\otimes^+_{2n_1+n_2-1}(T)$ is given by

$$\begin{aligned} \otimes^+_{2n_1+n_2-1}(T) = &\{b^i c^j \# \ : \ i \geq 0, j \geq 2n_1 + n_2\} \\ &\cup \{\$a^i b^j c^k \# : \ i,j,k \geq 2n_1 + n_2, i < j\}. \end{aligned}$$

Thus, the result holds by Lemma 14. □

We now turn to undecidability. As expected, we can not determine whether a linear context-free set of templates is minimal.

Theorem 18. *It is undecidable whether a linear context-free set of templates is $(1,1)$-minimal.*

Proof. Let $L_1, L_2 \subseteq \Sigma^*$ be arbitrary linear context-free languages, $\Delta = \Sigma \cup \{\#, \$\}$ and $T \subseteq \Delta^*$ be defined by

$$T = \$\#\#L_1\$ \cup \#\#L_2\$ \cup \{\$\#\#\}.$$

We claim that the set of templates T is $(1,1)$-minimal if and only if $L_1 \cap L_2 \neq \emptyset$.

If the intersection $L_1 \cap L_2$ is non-empty, let $x \in L_1 \cap L_2$ and note that $\$\#\#x\$ \in \$\#\# \otimes_2 \#\#x\$$, and so $\$\#\#x\$ \in T - \beta_{1,1}(T)$. On the other hand, suppose that there exists $x \in T - \beta_{1,1}(T)$. Then $x \in \otimes_2^+(T)$. By counting occurrences of the symbols \$ and #, we can see that the only way to construct a word in T by overlapping two or more words in T is by choosing \$## and ##\$ for some $y \in L_2$, to yield $\$\#\#y\$$. But this word must be in L_1, so $y \in L_1 \cap L_2$.

This establishes the claim and thus, since determining whether two linear context-free grammars have a non-trivial intersection is undecidable, the result follows. □

By direct simulation of (C1) with an LBA or by establishing closure properties of the context-sensitive languages under $\otimes_N^+$, we can easily establish the following result:

Lemma 19. *Let $n_1, n_2 \geq 1$ and let T be a context-sensitive set of templates. Then $\beta_{n_1,n_2}(T)$ is a context-sensitive set of templates.*

6. Conclusions

Template-guided recombination has a natural language-theoretic formulation which involves a language L, the set of DNA to be rearranged and T, the set of templates. The set of templates dictates which rearrangements on the words of L are possible.

In the context of natural computing, it is natural to consider the impact of modifications to the set of templates on the rearrangement process. These considerations yield the following equivalence problem: given two sets of templates T_1 and T_2, do they define the same formal language operation? This topic, including characterizations and decidability, has been examined in a previous paper by the author [10].

In this paper, we have examined the concept of minimality of sets of templates. Alongside the concept of equivalence, it is natural to consider the minimal (with respect to inclusion) set of templates which is equivalent to a given set T. By using previous results on equivalence [10], we have shown that $\beta_{n_1,n_2}(T)$, the minimal set of templates equivalent to T, is unique.

We have also shown that by rephrasing the original results on equivalence for sets of templates in terms of overlap operations on formal languages, we can obtain a operational description of equivalence. This operational description yields an explicit formula for constructing $\beta_{n_1,n_2}(T)$ from T, using the overlap operation $\otimes_N$.

With our formula for the minimal set, we obtain decidability results and closure properties. In particular, since the minimal set $\beta_{n_1,n_2}(T)$ for a regular set of templates T is again regular, we can effectively construct it, and determine whether a regular set of templates T is minimal. For context-free sets of templates, the set $\beta_{n_1,n_2}(T)$ is not necessarily context-free, and it is undecidable in general if $T = \beta_{n_1,n_2}(T)$.

The concept of minimality has obvious motivations in the application of DNA rearrangement to molecular computing. The use of the set of templates as controls on the allowed rearrangement is an important concept in viewing a ciliate as a programmable natural computer. The recent experimental work of Nowacki et al. [17] suggests that modifications to the set of templates is possible. Thus, it is natural to consider what is the minimal set of templates necessary to perform a certain rearrangement in general.

We note that the iterated version of TGR [6] has been the subject of much research, and is biologically motivated. However, the question of equivalence for iterated TGR is open [10]. Minimality for iterated TGR is also an important topic worth considering.

Acknowledgments

Many thanks to the referees for useful suggestions and comments.

References

[1] A. Alhazov, I. Petre, V. Rogojin, Solutions to computational problems through gene assembly, Natural Computing, 7 (2008) 385–401.
[2] A. Angeleska, N. Jonoska, M. Saito, L. Landweber, RNA-guided DNA assembly, Journal of Theoretical Biology 248 (2007) 706–720.
[3] A. Cărăuşu, G. Păun, String intersection and short concatenation, Revue Roumaine de Mathématiques Pures et Appliquées 26 (1981) 713–726.
[4] M. Daley, M. Domaratzki, Template-Guided Recombination: From Theory to Laboratory, Algorithmic Bio-Processes, Natural Computing Series, Springer, 2009.
[5] M. Daley, M. Domaratzki, A. Morris, Intra-molecular template-guided recombination, International Journal of Foundations of Computer Science 18 (2007) 1177–1186.
[6] M. Daley, I. McQuillan, Template-guided DNA recombination, Theoretical Computer Science 330 (2005) 237–250.
[7] M. Daley, I. McQuillan, On computational properties of template-guided DNA recombination in ciliates, in: A. Carbone, N. Pierce (Eds.), DNA Computing, Lecture Notes in Computer Science, vol. 3892, Springer-Verlag, 2006, pp. 27–37.
[8] M. Daley, I. McQuillan, Useful templates and iterated template-guided DNA recombination in ciliates, Theory of Computing Systems 39 (2006) 619–633.
[9] M. Domaratzki, Semantic shuffle on and deletion along trajectories, in: C. Calude, E. Calude, M. Dineen (Eds.), Developments in Language Theory, Lecture Notes in Computer Science, vol. 3340, Springer, 2004, pp. 163–174.
[10] M. Domaratzki, Equivalence in template-guided recombination, Natural Computing 7 (2008) 439–449.
[11] A. Ehrenfeucht, T. Harju, I. Petre, D. Prescott, G. Rozenberg, Computation in Living Cells: Gene Assembly in Ciliates, Springer-Verlag, 2004
[12] A. Ehrenfeucht, G. Rozenberg, Covers from templates, International Journal of Foundations of Computer Science 17 (2006) 475–488.

[13] M. Ito, G. Lischke, Generalized periodicity and primitivity, Mathematical Logic Quarterly 53 (2007) 91–106.
[14] A. Mateescu, G. Păun, G. Rozenberg, A. Salomaa, Simple splicing systems, Discrete Applied Mathematics 84 (1998) 145–162.
[15] A. Mateescu, A. Salomaa, Parallel composition of words with re-entrant symbols, Analele Universităţii Bucureşti Matematică-Informatică 45 (1996) 71–80.
[16] I. McQuillan, K. Salomaa, M. Daley, Iterated TGR languages: membership problem and effective closure properties, in: D. Chen, D. Lee (Eds.), Computing and Combinatorics, Lecture Notes in Computer Science, vol. 4112, Springer-Verlag, 2006, pp. 94–103.
[17] M. Nowacki, V. Vijayan, Y. Zhou, K. Schotanus, T. Doak, L. Landweber, RNA-mediated epigenetic programming of a genome-rearrangement pathway, Nature 451 (2008) 153–159.
[18] I. Petre, V. Rogojin, Decision problems for shuffled genes, Information and Computation 206 (2008) 1346–1352.
[19] E. Post, Formal reductions of the general combinatorial decision problem, American Journal of Mathematics 65 (1943) 197–215.
[20] D. Prescott, A. Ehrenfeucht, G. Rozenberg, Template-guided recombination for IES elimination and unscrambling of genes in stichotrichous ciliates, Journal of Theoretical Biology 222 (2003) 323–330.
[21] G. Rozenberg, A. Salomaa (Eds.), Handbook of Formal Languages, Springer-Verlag, 1997.